

Applicant : Gilbert Wolrich et al.
Serial No. : 10/069,229
Filed : December 11, 2002
Page : 2 of 8

Attorney's Docket No.: 10559-305US1 / P9626US

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently amended) A computer instruction, comprises:
a branch instruction that causes a processor to branch from executing a first sequential series of instructions to a different sequential series of instructions based on a byte in a register being equal or not equal to a specified specified byte value, if the specified byte matches or mismatches the byte value.
2. (Original) The instruction of claim 1 wherein the branch is to an instruction at a specified label.
3. (Original) The instruction of claim 1 wherein the branch instruction comprises:
a bit_positon field that specifies the byte in a longword contained in the register.
4. (Original) The instruction of claim 1 wherein the branch instruction comprises:
an optional token that is set by a programmer and specifies a number i of instructions to execute following the branch instruction before performing the branch operation.
5. (Original) The instruction of claim 1 wherein the branch instruction comprises:
an optional token that is set by a programmer and specifies a number i of instructions to execute following the branch instruction before performing the branch operation where the number of instructions can be specified as one, two or three.

Applicant : Gilbert Wolrich et al.
Serial No. : 10/069,229
Filed : December 11, 2002
Page : 3 of 8

Attorney's Docket No.: 10559-305US1 / P9626US

6. (Original) The instruction of claim 1 wherein the register is a context-relative transfer register or a general-purpose register that holds the operand.

7. (Original) The instruction of claim 1 wherein the branch instruction comprises: an optional token that is set by a programmer and which specifies a guess_branch prefetch for the instruction for the "branch taken" condition rather than the next sequential instruction.

8. (Original) The instruction of claim 1 wherein the branch instruction comprises: an optional token that is set by a programmer and specifies a number i of instructions to execute following the branch instruction before performing the branch operation; and a second optional token that is set by a programmer and which specifies a guess_branch prefetch for the instruction for the "branch taken" condition rather than the next sequential instruction.

9. (Original) The instruction of claim 1 wherein the branch instruction allows a programmer to which bit of the register to use to determine the branch operation.

10. (Original) The instruction of claim 1 wherein the branch instructions allows branches to occur based on evaluation of a byte that is in a data path of a processor.

11. (Original) The instruction of claim 1 wherein the branch instruction branches on a byte matching the byte value and wherein the instruction prefetches the instruction for the "branch taken" condition.

12. (Original) The instruction of claim 1 wherein the branch instruction branches on a byte not matching the byte value and wherein the instruction prefetches the next sequential instruction.

Applicant : Gilbert Wolrich et al.
Serial No. : 10/069,229
Filed : December 11, 2002
Page : 4 of 8

Attorney's Docket No.: 10559-305US1 / P9626US

13. (Original) The instruction of claim 1 wherein the branch instruction includes a Byte_spec Number that specifies the byte in the register to be compared with byte_compare_value.

14. (Currently amended) A computer program product residing on a computer readable medium for causing a processor that executes multiple contexts to perform a function comprises instructions causing the processor to:

fetch a byte stored in a register;
determine whether the byte in the register is equal or not equal to a specified specified byte value contained in the instruction; and
perform a branching operation specified by the branch instruction based on the specified byte being equal or not equal to the byte in the register

15. (Original) The product of claim 14 wherein the branch is to an instruction at a specified label.

16. (Original) The product of claim 14 wherein the program includes a branch instruction that comprises:

a bit_positon field that specifies the byte in a longword contained in the register.

17. (Currently amended) A processor comprises:
a register stack;
an arithmetic logic unit coupled to the register stack and a program control store that stores a branch instruction that causes the processor to:

fetch a byte stored in a register;
determine whether the byte in the register is equal or not equal to a specified specified byte value contained in the instruction; and

Applicant : Gilbert Wolrich et al.
Serial No. : 10/069,229
Filed : December 11, 2002
Page : 5 of 8

Attorney's Docket No.: 10559-305US1 / P9626US

perform a branching operation specified by the branch instruction based on the specified byte being equal or not equal to the byte in the register.

18. (Original) The processor of claim 17 wherein instructions to perform the branch branch to an instruction at a specified label.

19. (Original) The processor of claim 1 wherein a bit_positon field in an instruction specifies the byte in a longword contained in the register.

20. (Currently amended) A method of operating a processor comprises:
executing a branch instruction by:
fetching a byte stored in a register;
determining whether the byte in the register is equal or not equal to a specified specified byte value contained in the branch instruction; and
performing a branching operation specified by the branch instruction based on the specified byte being equal or not equal to the byte in the register.

21. (Original) The method of claim 20 wherein performing the branch branches to an instruction at a specified label.